

# 1 - Introduction

---

## 21S1

---

**Q1: Discuss the key challenges that one needs to address in the design and development of distributed systems or applications.**

- Heterogeneity
  - Heterogeneous components must be able to interoperate.
  - The system should be compatible to different software and hardware.
- Distribution transparency
  - Distribution should be hidden from the user as much as possible.
  - Users should have the same experience in different locations.
- Fault tolerance
  - Failure of a component (partial failure) should not result in failure of the whole system.
- Scalability
  - Work efficiently with increase of users.
  - Performance should increase with inclusion of resources.
- Concurrency
  - Shared access to resources must be possible.
  - Avoid deadlocks.
- Openness
  - Interfaces should be publicly available to ease inclusion of new components.
  - APIs are provided.
- Security
  - The system should only be used in the way intended.

**Q12.1: The “Openness” challenge of Distributed System means the following:**

- Ability to easily open source the code base
- Ability to easily extend the existing system
- Ability to change the current system
- Ability to make system accessible to all ✓

## 17S1

---

**Q1: (a) [5 marks]** Consider a non-distributed application that allows a user to conduct a long-running simulation, with the ability to monitor and control the simulation using a graphical user (GUI) interface. The GUI interface displays a large number of simulation outputs in real-time. Now consider the same application that has been split into two processes: *a server process that runs the simulation, and a GUI process that allows user monitoring and control*. This is now a distributed application. Give three reasons for and two reasons against the use of a distributed application with respect to this example. Make sure your reasons are all distinctly different.

Advantage: P4-7

- Functional Separation
  - Simulation & GUI can be separated to increase the scalability of each part. For example, the simulation part can be scaled up, ignoring the GUI part.
- Reduce the load of a single server
  - No need to process both simulation and GUI tasks simultaneously. (Parallel Activities)
- Increase availability and reliability
  - If the GUI part fails, the simulation part can still work. (Independent failures)

Disadvantage: P8

- More difficult to maintain since there is an extra part.
- These two components must cooperate with each other to solve heterogeneity.

**(b) Explain what is meant by eventual consistency in a distributed system. You may give an example application to aid your answer. Give your reasons for and against adopting eventual consistency in a distributed system.**

**Eventual consistency** is a consistency model used in distributed computing to achieve high availability that informally guarantees that, if no new updates are made to a given data item, eventually all accesses to that item will return the last updated value. (Wikipedia)

For example, every one can post tweets on twitter. It may take some time to synchronize all the new tweets, but if no update is made for a while, users all over the world should eventually see the same tweets.

Advantage:

Reduce the load of synchronization and save the cost of databases. Good if the amount of users is huge.

Disadvantage:

It is a form of weak consistency so that users may have different experiences if the synchronization is not finished yet.

**Q1: (a) [5 marks] Consider a large computer lab, e.g. like those found on a university campus. The computer lab allows users to login to any computer using their username and password and makes various software and other resources available that are specific to that user. List and briefly explain, with respect to this example, five challenges of distributed systems.**

Challenges

**Q3: (a) [3 marks] Briefly explain three fundamental aspects of a communication channel that are important from the point of view of a distributed system. P7**

**Q6: (b) [1 marks] Explain the difference between virtualization and emulation.**

L1 P12

Virtualization: Managing access to existing physical devices such that there appears to be more of them or greater capacity from a software perspective (i.e. logically), essentially time sharing

Emulation: Create the appearance of a device existing, even though it does not

## 15S2

---

**Q1: (a) [1 marks] Provide a clear definition of independent failure in a distributed system.**

An independent failure is **a malfunction of a system component that does not affect any other component in the system.**